

PROGRAMMABLE LOGIC DEVICE WITH METHOD OF PRREVENTING READBACK

Publication number: WO0244875 (A2)

Publication date: 2002-06-06

Inventor(s): PANG RAYMOND C; SZE WALTER N; THENDEAN JOHN M; TRIMBERGER STEPHEN M; WONG JENNIFER

Applicant(s): XILINX INC [US]

Classification:

- international: H03K19/173; G06F17/50; G06F21/00; G09C1/00; H04L9/10;
H03K19/173; G06F17/50; G06F21/00; G09C1/00; H04L9/10;
(IPC1-7): G06F1/00

- **European:** G06F17/50D4; G06F21/00N1C5

Application number: WO2001US45055 20011128


Priority number(s): US20000724975 20001128

Also published as:

WO0244875 (A3)

US6981153 (B1)

JP2004515180 (T)

 EP1358530 (A2)

CA2429597 (A1)

Cited documents:

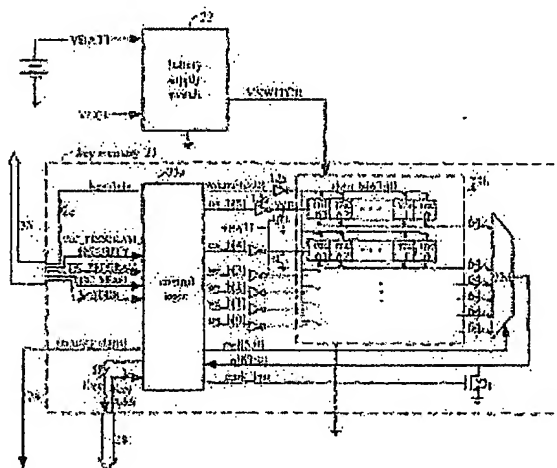
 US5970142 (A)

US2001015919 (A1)

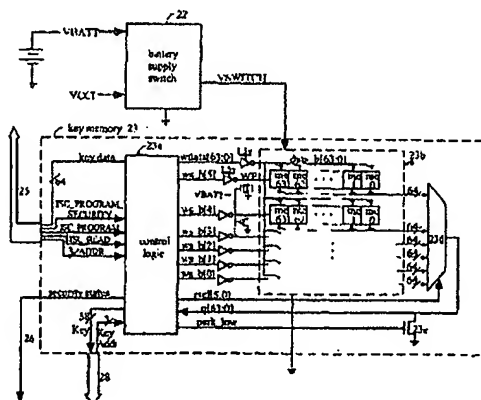
WO0049717 (A2)

Abstract of WO 0244875 (A2)

It is sometimes desirable to protect a design used in a PLD from being copied. If the design is stored in a different device from the PLD and read into the PLD through a bitstream, the design may be encrypted as it is read into the PLD and decrypted within the PLD before being loaded into configuration memory cells for configuring the PLD. According to the invention, in such a device, a method is provided to prevent the design from being read back from the PLD in its decrypted state if it had been encrypted when loaded into the PLD.



Data supplied from the *esp@cenet* database — Worldwide



【特許請求の範囲】

【請求項 1】

PLDを構成するための設計を保護するための方法であって、
PLD内に、設計を表現する暗号化されたビットストリームをロードするステップと、
前記PLD内のビットストリームを復号して、前記設計を表現する暗号化されていないビットストリームを生成するステップと、
暗号化されていない前記設計で前記PLDを構成するステップと、
前記暗号化されていないビットストリームのリードバックを禁止するステップとを含む、方法。

【請求項 2】

PLDを構成するための設計を保護する方法であって、
PLD内に、設計を表現する暗号化されたビットストリームをロードするステップと、
前記PLD内のビットストリームを復号して、前記設計を表現する暗号化されていないビットストリームを生成するステップと、
前記暗号化されていないビットストリームで前記PLDを構成するステップと、
前記PLDが部分的に再構成されることを禁止するステップとを含む、方法。

【請求項 3】

前記PLDが部分的に再構成されることを禁止する前記ステップはさらに、ユーザの設計の或る部分が前記PLDの別の部分に再配置されることを防ぐステップを含む、請求項 2 に記載のPLD。

【発明の詳細な説明】

【0001】

【発明の分野】

この発明はPLDに関し、より特定の、ビットストリームを介してPLD内にロードされる設計の保護に関する。

【0002】

【発明の背景】

PLD（プログラマブルロジックデバイス）とは、設計者が選択したデジタル論理関数を実行する集積回路構造である。PLDはロジックブロックと相互接続線とを含み、典型的にロジックブロックおよび相互接続はともにプログラム可能である。1つの一般的な種類のPLDがFPGA（フィールドプログラマブルロジックデバイス）であり、ここでロジックブロックは典型的にルックアップ表およびフリップフロップを含み、典型的にその入力信号のあらゆる関数を生成および記憶できる。別の種類はCPLD（コンプレックスプログラマブルロジックデバイス）であり、ここではロジックブロックはAND関数、およびOR関数を実行し、入力信号の選択はプログラム可能である。

【0003】

PLDの外部でのビットストリームの記憶に関する問題

PLD内で実現される設計は複雑化しており、PLD内で実現すべき設計を完成させデバッグするにはしばしば数ヵ月を要する。PLDの一部をなすシステム内に設計が入れられ、利益のため販売される際、設計者にとって誰か他の者がこの設計努力をコピーするという結果は望ましいものではない。設計者はしばしば設計を企業秘密にしておこうと望む。多くのPLD、特にFPGAは揮発性構成メモリを使用し、これにはPLDに電源が投入されるたびにPROMなど外部のデバイスからロードされることになる。構成データはPLDの外部に記憶され、構成アクセスポートを通じて伝送されることになるため、攻撃者、すなわち構成アクセスポート上のデータをたとえばボードトレース上にプローブを置くことにより監視する者によって、設計の機密は容易に侵される恐れがある。

【0004】

現在の解決策およびその欠点

設計を暗号化する試みがなされてきているが、設計を攻撃者に対して安全にし、かつ正規のユーザによる使用を容易にすることは困難である。暗号化アルゴリズムは問題ではない

10

20

30

40

50

。データブロックを暗号化するためには、いくつかの暗号化アルゴリズム、たとえば標準的なデータ暗号規格 (DES) およびより安全な高度暗号規格 (AES) のアルゴリズムなどが公知である。暗号ブロック連鎖 (CBC) のプロセスでは、暗号化されていないデータワードと次の暗号化されたデータワードとの XOR をとってから復号が行われ、これによって DES または AES はデータのシリアルストリームを暗号化できるので、PLD を構成するためのビットストリームを暗号化するのに好適である。設計を暗号化するのに用いる鍵は何らかの形で、PLD と、設計を復号する構造との間で安全なやり方で通信されることになり、こうして設計が PLD により復号され、PLD を構成するのに用いられ得る。次に、暗号化されていない設計を用いて PLD が一旦構成されると、設計を無認可で発見されないよう保護し続けなければならない。

10

【0005】

「構成に関する問題：電源投入、揮発性、安全性、電池のバックアップ (Configuration Issues : Power-up, Volatility, Security, Battery Back-up)」と題された、ザイリンクス・インコーポレイテッド (Xilinx, Inc.) のピーター・アルフキー (Peter Alfke) による 1997 年 11 月 24 日の刊行物には、設計を保護するために、FPGA 内に特定のアーキテクチャ上の特徴を有さない、既存の FPGA デバイスでの設計を保護するためにとられ得る、いくつかのステップが記載されている。1つの方法は、FPGA 内に設計構成データをロードしてから構成データのソースを外すが、電池を用いて FPGA に対して連続的に電力を維持し、一方で FPGA を待機非動作モードに保つことである。しかし電池に対する電力の要件のため、この方法は大規模な FPGA デバイスにとって非実用的なものとなっている。

20

【0006】

別の可能性は不揮発性構成メモリである。デバイスの販売前に設計が工場ロードされる場合、構成された PLD デバイスの購入者にとって、設計がどのようなものであるかを判断することは困難である。しかし逆行分析によって、プログラムされたデバイスの覆いを取去り、金属層を除去し、不揮発性メモリセルに化学処理を施すと、どのメモリセルが充電されたかが明らかとなる可能性があり、こうして攻撃者は設計を知ることができる恐れがある。さらに不揮発性メモリには、標準的な CMOS プロセス技術よりも複雑で経費のかかるプロセス技術が必要であり、市場に出るまでにより多くの時間がかかってしまう。

30

【0007】

PLD の不揮発性メモリ内に復号鍵を記憶させ、暗号化されたビットストリームを PLD 内にロードし、PLD 内の鍵を用いてビットストリームを復号することもまた公知である。こうすれば、攻撃者がビットストリームを PLD 内へのロード中に読出すことが防止され、かつ、電力が PLD から外されても鍵が維持される。このような機構は、オースティン (Austin) による米国特許第 5,388,157 号に記載されている。しかしこの構造はユーザの設計をあらゆる態様の攻撃から保護するわけではない。

【0008】

設計保護に加え、ユーザによってはデータ保護もまた必要である。PLD が電力を失っても失われてはならないデータを、ユーザが PLD 内に生成することがある。このようなデータは保護されることが望ましい。

40

【0009】

したがって、便利で信頼性が高く、かつ安全な設計保護方法に対する必要性がなお存在している。

【0010】

【発明の概要】

この発明は、PLD を無認可の使用およびデータの紛失から保護するためのいくつかの構造および方法を提供する。

【0011】

電源投入の際にロードされなければならない静的 RAM メモリによって PLD が構成され

50

る場合、構成データはデバイスへのロードの際に保護されなければならない。先行技術においてこれは、構成データを暗号化して集積回路デバイスの外部にあるメモリに記憶させ、1つ以上の復号鍵をPLD内にロードして電源切断時に鍵をPLD内に維持し、鍵を用いて構成データを復号する復号回路をPLD内に含め、復号された構成データをPLD内で生成し、復号された構成データを用いてPLDを構成することによって達成される。

【0012】

さらなる安全性のために、不揮発性メモリを用いて鍵を保存する代わりに、この発明は好ましくはPLDに接続された電池を用いて、PLDから電力が外される際に鍵を保存する。不揮発性メモリ内に鍵を記憶させたPLDを取外し、PLDの覆いを外し、不揮発性ビットのうちどれが論理1にプログラムされどれが論理0にプログラムされているかを観察することは可能であるが、静的メモリセルにのみ記憶された鍵の内容を判断することは極めて困難であり、なぜなら、鍵を記憶させるためだけでも鍵を記憶するメモリセルには電力が維持されなければならない、連続的にPLDに対し動作電力がある間にPLDの覆いを外し、層を取除き、プローブをかけなければならないであろうからである。

【0013】

一旦PLD内にロードされた設計を攻撃者が盗み得るやり方

鍵が十分な安全性を提供していなければ、攻撃者は暗号化コードを破って鍵の値を判断する可能性がある。周知のデータ暗号規格DESが用いていた暗号鍵は56ビットのものであり、鍵解読用の高性能のコンピュータによって数時間で破られてしまった。DESはブルース・シュナイヤー (Bruce Schneier) によって、ブルース・シュナイヤー著、ジョン・ワイリー&サンズ・インコーポレイテッド (John Wiley & Sons, Inc.) 刊行の「応用暗号技術第2版：プロトコル、アルゴリズム、およびCにおけるソースコード (Applied Cryptography Second Edition: protocols, algorithms, and source code in C)」著作権1996年の265～278頁で説明されている。このような周知の暗号規格を用いることが所望であれば、安全性を増大させるために構成データは数度、各回につき異なった鍵を用いて暗号化され、こうして暗号化を繰返すたびに約 2^{56} だけ暗号化コードを強化することがある。または構成データを、一番目の鍵を用いて暗号化し、2番目の鍵を用いて復号し、3番目の鍵を用いて暗号化することがあり、この組合せは三重DES規格の一部である。その他の暗号化アルゴリズムを用いることもできるが、安全性は鍵に存するためアルゴリズムを秘密にしておく必要はない。暗号化の方法が対称であれば、暗号化に用いた同じ鍵がPLD内に記憶され、復号では逆の順序で用いられる。

【0014】

多数の鍵を提供するPLDでは、用いるべき鍵の数とあらゆる鍵のアドレスとが、暗号化されていないビットストリームで提供される場合、攻撃者は1回につき1つずつ鍵に攻撃をかけて鍵の値をより容易に判断できる可能性がある。このような攻撃を回避するため、いくつかの鍵を用いるのか、および、鍵がセット内の最後の鍵か、またはさらなる鍵が続くのかについての指示を、ビットストリーム内でなく鍵内に記憶することによって追加の安全性を達成する。

【0015】

ビットストリームがPLD内にロードされた後にビットストリームをリードバックする選択肢をPLDが提供する場合、攻撃者が用い得る別の方法はこのビットストリームをリードバックすることである。設計を攻撃するこの方法を回避するため、一実施例で、リードバックを提供しかつ暗号化も提供するPLDは、暗号化が使用された場合にはリードバックのフィーチャを禁止する能力を含む。別の実施例では、リードバックの能力を提供するPLDは構成データをそのリードバック前に暗号化する。

【0016】

加えていくつかのPLDは、部分的構成 (いくつかの構成アドレスを特定して設計におけるいくつかの部分ロードする) および部分的再構成 (既存の設計が消去されずに新たな

10

20

30

40

50

設計データがロードされる)の選択肢を提供する。PLDがこれらの選択肢を提供する場合、攻撃者はPLDを部分的に再構成し、連続する設計部分を可視にして、おそらくは設計全体を知る可能性がある。このような攻撃を回避するため、一実施例では、暗号化された設計がロードされたPLDの部分的な構成および再構成は却下される。別の実施例では、いくつかの構成アドレスが特定され得るが、アドレスは暗号化される。

【0017】

さらに別の攻撃態様は、PLDの安全性状態を示すビットを反転させようとする試みである。動作電圧を低下または上昇させる、温度を変化させる、および或るポートに雑音を加えることが考えられる。このようなビット反転から保護するため、安全にされたビットストリームでPLDが動作しているときには安全モードのフラグをセットし、一実施例では、このフラグが非セットされるとすべての構成データが消去される。デバイスがまだ動作している間に再構成を許さない別の実施例では、構成データはビットストリームのいずれかが送信される前に消去される。

【0018】

別の攻撃態様は、暗号化されたビットストリームの或る部分を再配置することによって、暗号化されていないときにこれら部分が、設計者の意図していないPLDの可視部分に置かれるようにすることである。この再配置を防ぐため、暗号化および復号のプロセスでアドレス情報を使用することによって、設計者が意図したPLD場所とは異なるPLD場所に暗号化されたビットストリームの一部を送ると、意味のないデータへの異なる復号が行なわれるようにする。暗号ブロック連鎖(CBC)は、この結果を達成する1つの有効な手段である。暗号ブロック連鎖では、復号されたデータブロック(ブロック)が次のデータブロックとXOR関数を用いて組合されてから次のブロックが復号され、こうして各データブロックについての暗号化されたデータは、これに先行するあらゆるブロックと、これらブロックの順序とに依存する。同一のデータブロックは、これに先行するデータブロックの値に依存して異なった値へ暗号化される。こうして、ブロックの順序が変えられるとビットストリームは正しく復号しないが、それは暗号化されたビットストリームが再配列されるところがその後のデータをスクランブルするからである。さらに、初期CBC値に変更を加えてデータのアドレスを組込ませ、復号されたデータが、正しく復号するための特定の場所に置かれるよう強制することができる。

【0019】

これに代えて、設計の一部が暗号化されかつ別の部分が暗号化されないことをPLDが許す場合、攻撃者は暗号化されていない部分を暗号化された部分に加えることができ、これは設計の暗号化された部分についての情報を読み出すことになる。したがって、設計すべてが暗号化されているか、または設計がいずれも暗号化されていないかのいずれかを可能にし、混合をなくすことによって、追加の安全性が達成される。これに加えて一実施例では、データ暗号化の際に、構成データについての単一の開始アドレスに続く単一のフルチップ構成のみを許すことによって、追加の安全性がもたらされる。

【0020】

さらに、試験およびデバッグを手頃にし、かつPLD製造業者とその顧客(すなわちPLDを構成するための設計を生産する設計者)との自由な通信を可能にするために、PLDは暗号化される動作モードと暗号化されない動作モードとの両方を有し、暗号化されるモードで動作する際、PLD内への構成データのロードを制御する構成ビットストリームの部分はまだ暗号化されていない。

【0021】

別の攻撃態様として、PLD製造業者が、構成データをロードするためのアドレスおよびヘッダ情報を含む構成ビットストリームフォーマットについて自由に情報を提供、および使用される暗号化方法についての情報を提供する場合、この周知の情報を暗号化すると、暗号鍵があり得る発見に対して暴露されてしまうことになるだろう。このような暴露は、実際の構成データのみを暗号化し、制御情報を暗号化されないままにすることによって回避される。

10

20

30

40

50

【0022】

PLD製造業者が安全モードおよび非安全モードの両方で鍵メモリを用いることを許す場合、攻撃者は単に鍵メモリを非安全モードに置いて鍵を読出すことによって鍵を知ることができるであろう。このような攻撃を回避するため、PLD製造業者は、回路であって、鍵メモリが非安全モードへ動かされた際、すべての鍵と、PLD内にロードされたあらゆる構成データとの消去を引起こす回路を含める。

【0023】

[詳細な説明]

図1は先行技術のFPGA10の構造を示す。FPGAはプログラマブルロジック11を含み、これは典型的に(1)ルックアップ表の組合せ論理関数生成器を伴ったロジックブロックと、ルックアップ表出力およびその他の値を記憶するためのフリップフロップと、プログラマブルロジックの論理能力を増大させるための論理ゲートおよびマルチプレクサを含み、さらに(2)FPGAのまわりに信号を経路付けするための経路付け線およびプログラム可能相互接続点と、(3)FPGAの外部ピンおよび経路付け線間で信号を駆動するための入出力ブロックとを含む。

10

【0024】

FPGAはさらに構成メモリ12を含み、これは経路付けトランジスタをオンにし、マルチプレクサを制御し、ルックアップ表を記憶し、さらに入出力ブロックを制御するが、これらすべてはFPGAを構成して開発者の所望の関数を実行することを目的としたものである。バス16は構成メモリ12をプログラマブルロジック11に接続し、典型的にFPGA全体にわたって位置付けられた分散した1組の制御線である。いくつかのザイリンクス製品(たとえばXC6200)はバス17を含んでおり、これによってプログラマブルロジック11は、構成ロジック14がプログラミング情報を構成メモリ12に送ることを引起こす。このような構造はキーン(Keen)による米国特許第5,705,938号で説明されている。

20

【0025】

FPGA10はさらに、JTAGポート20とインターフェイスするためのJTAGロジックブロック13を含み、これは特に、FPGAが置かれることになるボードの試験を意図したものである。JTAGロジックブロック13はIEEE規格1532を満たすが、これはIEEE規格1149.1の上位セットである。JTAGはボードレベルでの設計のデバッグを可能にする。

30

【0026】

最後に、FPGA10は構成ロジック14を含み、これは構成アクセスポート21上の、外部ソース15からの構成ビットストリームに応答し、さらにJTAGロジックブロック13にインターフェイスする。構成アクセスポート21上のビットストリームはワード、一実施例では32ビットワードとして扱われる。通常はビットストリームの初めまたはその近くにあるワードのいくつかは構成プロセスを準備するために用いられ、たとえば構成メモリフレームの長さ、および構成データについての開始アドレスを含む。バス19は構成ロジック14とJTAGロジックブロック13との間の通信を可能にし、こうしてJTAGポートはもう1つの構成アクセスポートとして使用可能である。バス18は構成ロジックブロック14と構成メモリ12との間の通信を可能にする。具体的には、メモリ12内で構成フレームを選択するためのアドレスと、読み書き動作を行なうための制御信号と、構成メモリ12内にロードまたは構成メモリ12からリードバックするためのデータとを伝える。

40

【0027】

構成ロジックブロック14は命令およびデータを受取り、命令に従ってデータを処理する。これら命令はビットストリームとして構成ロジック14に入る。命令、またはヘッダには通常、処理を受けることになるデータが続く。図2aは例示的なビットストリーム構造を示す。ヘッダAはアクションを特定し、単一のワード、すなわちデータAが続くことを特定する。ヘッダBはアクションを特定し、この場合は4ワードのデータが続いて処理を

50

受けることを特定する。

【0028】

図2bは、ザイリンクス・インコーポレイテッドから入手可能なパーテックス(R) (V i r t e x) デバイスで用いられるビットストリーム内の32ビットヘッダワードについてのデフォルトフォーマット(フォーマットタイプ001)を示す(パーテックスは、この発明の譲受人であるザイリンクス・インコーポレイテッドの登録商標である)。このフォーマットは、フォーマットタイプ(001)を示すための3つのビットと、オペコードを特定するための2つのビットと、構成ロジックレジスタアドレスのための16ビットと、ワードカウントのための11ビットとを含む。オペコードは読出動作、書込動作または動作なしを指定し得る。たとえば00は動作なしを指定し、01は読出を指定し、10は書込を指定することがあり得る。ワードカウントのための11ビットは 2^{11} ワードまたは2048ワードを特定し得る。図2cに示すように、ワードカウントがこれよりも大きければ、フォーマットタイプ001におけるワードカウントビットは00000000000に設定され、フォーマットタイプ001のヘッダにはフォーマットタイプ2のヘッダが続く。フォーマットタイプ2はワードカウントを特定するために27ビットを使用し、こうして 2^{27} ワードまたは268万ワードを特定し得る。

10

【0029】

図2dは、パーテックスのビットストリームについてのヘッダにより構成ロジック14のレジスタ内にロードされ得るような制御情報を示す。たとえば、構成ロジックレジスタアドレス0000を有する(フォーマット001の)ヘッダは、次の32ビットデータワードが巡回冗長検査(CRC)レジスタ内にロードされるべきであることを特定する。(パーテックスのデバイスは16ビット巡回冗長検査値を用いるため、いくつかのビットは0で埋められる。)ヘッダがアドレス0001を含む場合、次のデータはフレームアドレスレジスタ内にロードされて構成メモリ12内でフレーム(列)を特定し、こうしてデータが受取られるまたは供給されることになる。

20

【0030】

図2bに示す構成ロジックレジスタアドレス(16ビット)は、図2dの左の列に示す4ビットの値を与え、これら値は、次の32ビットデータワードを置くべき構成ロジック14(図1)内のレジスタの1つを選択する。フレーム長レジスタ(アドレス1011)は、構成データをロードすることになるフレームの長さを特定する。(フレーム長または列の高さはPLDのサイズに依存する。通常、より大きいPLDはより高い列またはより長いフレームを有する。PLD内に異なる構造を設けてデータワードをフレーム内に置く代わりに、ビットストリーム内でフレーム長を特定しレジスタ内でフレーム長を記憶することにより、異なったサイズのPLDについて内部構成ロジックが同一となることができる。)

30

【0031】

リードバックのためには、読出コマンドがオペコードフィールド内に置かれ、フレームデータ出力レジスタがアドレス指定され、これに(必要であればコマンドヘッダフォーマット2を用いた)ワードカウントが続く。特定された数のワードは、フレームアドレスレジスタ内で特定されたアドレスから始まって構成メモリ12からリードバックされ、構成アクセスポート21またはJTAGポート20へシフトされる(リードバックデータは、リードバック命令を発行したポートへ戻される)。

40

【0032】

ビットストリームヘッダまたはヘッダ対(図2bおよび図2c)でワードカウントが特定されるとカウンタがセットされ、これはデータワードがロードされるのに伴ってカウンタダウンする。多くの構成ロジックレジスタアドレスではワードカウントは1である。しかし、構成データがロードまたはリードバックされていることを示すための0010または0011の構成ロジックアドレスをビットストリームヘッダが有する場合、ワードカウントははるかに大きくなるであろう。これは図2cのヘッダフォーマット2を使用する場合である。フレームデータ入力レジスタ(アドレス0010)を通じて構成メモリ12内に

50

ロードされたデータ、またはフレームデータ出力レジスタ（アドレス0011）を通じてリードアウトされたデータは設計データと呼ばれるが、それはこのデータがFPGAに設計を実現させる、または設計の状態を示すからである。他のレジスタデータは制御データであるが、それはこれらデータによって、ロジックの構成またはリードバック中に構成ロジックがどう振舞うかが制御されるからである。

【0033】

バーテックスのデバイスの構成についてのさらなる詳細は、ザイリンクス・インコーポレイテッド（この発明の譲受人）、95124 カリフォルニア州、サン・ノゼ、ロジック・ドライブ（Logic Drive, San Jose, CA）、2100による、2000年10月9日刊行の「バーテックス構成ガイド（Virtex Configuration Guide）」に記載されている。

10

【0034】

構成ロジック14は典型的に、入って来る構成ビットストリームに対して巡回冗長検査（エリクソン（Ericsson）による米国特許第5,321,704号を参照、または先に参照したバーテックス構成ガイドの39頁から40頁を参照）を行ない、構成されている部分のフレーム長と構成データのワードカウントとを示すヘッダビットを読み出し、どこに構成データをロードすべきかを同定するアドレス命令を読み出し、構成データのフレームを収集し、アドレスに示された構成メモリ12の列にこれらフレームをロードする。構成ロジック14はさらに構成メモリ12から外部の場所への構成データおよびフリップフロップ値のリードバックを制御する。ザイリンクス・インコーポレイテッドから入手可能なバーテックスFPGAでは、JTAGポート20または構成アクセスポート21を通じてリードバックが行なわれ得る。

20

【0035】

構成ロジック14はさらにプログラマブルロジック11から構成データを受取り得る。FPGAの或る部分がFPGAの別の部分を構成する先行技術のFPGA構造についてのさらなる情報は、キーンによる米国特許第5,705,938号に記載されている。バーテックスのアーキテクチャに類似のFPGAアーキテクチャについてのさらなる情報は、ヤング（Young）他による米国特許第5,914,616号に記載されている。この発明の譲受人であるザイリンクス・インコーポレイテッドから入手可能なバーテックス製品で用いられるビットストリームのフォーマットは、ザイリンクス・インコーポレイテッド、95124 カリフォルニア州、サン・ノゼ、ロジックドライブ、2100から入手可能である、2000年10月4日刊行の「バーテックスFPGAシリーズ構成およびリードバック（Virtex FPGA Series Configuration and Readback）」と題された出願ノートXAPP138に記載されている。

30

【0036】

復号を伴うPLD

図3は、この発明の一実施例に従うFPGA（PLDの1種）のブロック図を示す。いくつかの要素は図1に示すものと同じであり、これらには同じ参照番号を付し、もはや説明は行なわない。これに加えて図3は、拡張された構成ロジックユニット29、復号器24および鍵メモリ23を含む。図3は、鍵メモリ23がJTAGアクセスポート20からバス25によってロードされる実施例を示す。実施例によっては、鍵メモリ23には別のポートを通じてロードされる。バス25は、データ、アドレス、および読み書き動作を行なうための制御信号を伝え、JTAGポート20からの復号鍵のプログラミングを可能にする。一実施例では、バス26が構成ポートからの鍵のプログラミングを可能にする。別の実施例ではバス26はなくされる。さらに別の実施例では、バス26はあり、バス25はなくされる。ここでさらに説明する実施例では、バス26は安全性に関するデータを鍵メモリ23から構成ロジック29に伝える。一実施例では、バス27は暗号化された構成データを構成ロジック29から復号器24に伝え、復号された構成データを構成ロジック29に戻す。バス28は復号器24がデータの復号のために鍵にアクセスすることを可能にする。暗号化されたデータが図3の構造にロードされる際、ロード中のビットストリーム

40

50

を監視する攻撃者は暗号化されたビットストリームのみを受取るため、この方法でユーザの設計を知ることはできない。

【0037】

部分的に暗号化されたビットストリーム

この発明の別の局面に従うと、ビットストリームは2つの部分、すなわち暗号化され得るまたはされ得ないユーザの設計を表現するデータ部分と、ビットストリームのロードを制御する制御部分とを含む（たとえば連続するビットストリーム部分がロードされるべきPLD内の列アドレスを与え、ロード動作の信頼性を調べるための巡回冗長検査（CRC）コードと、暗号ブロック連鎖（CBC）のためのスタータ値とを提供するが、この技術は、暗号化されたデータの発生の頻度から復号データが推理され得る「辞書攻撃」を防止する）。この発明の好ましい実施例では、データ部分は暗号化され得るが、制御部分は暗号化されない。これによって追加の安全性がもたらされるが、それはPLD製造業者はビットストリームの制御フィーチャを自由に記述する必要があり、もしこの比較的よく知られた制御情報が暗号化されれば、攻撃者はこの情報を解釈し、この情報を用いてビットストリーム全体を解釈できるというおそれがあるからである。さらに、ビットストリームの制御部分を暗号化されないままにしておくことによって、PLDによる情報の使用がより容易となる。

10

【0038】

構成データがロードされるアドレスの順序が攻撃者にとって設計の分析に有用となり得る場合に用いられる別の実施例では、構成データのアドレスもまた暗号化されるが、構成ビットストリームにある他の制御情報は暗号化されないままである。

20

【0039】

ビットストリームフォーマット

図4a～4dは、図2a～2dに示す先行技術の製品の構成ロジック14のレジスタおよびビットストリームフォーマットと比較した場合の、構成ロジック29のレジスタおよびビットストリームフォーマットにおける違いを例示する。図4aに示すように、ビットストリームはやはりヘッダワードを含み、これにデータワードが続く。典型的な構成では、いくつかの制御データワードは、暗号化された構成データが始まる前にレジスタ内にロードされる。図4aは、3つのヘッダワード、すなわちヘッダA、ヘッダBおよびヘッダCに各々3つの暗号化されない制御データワード、すなわちデータA、データBおよびデータCが続く例を示す。（実際の構成では、3つを上回る制御データワードが与えられるであろう。）次にヘッダDは、暗号化された構成データが続くであろうことを特定し、多数のワード、すなわち暗号化された構成データであるデータ1D、データ2D、データ3Dなどが続く。図4aではこれらワードに影をつけ、このデータが暗号化されていることを強調する。

30

【0040】

図4bおよび図4cに示すように、第4のオペコードが追加されている。動作なしについての値00、復号のない読み書きについての01および10に追加された新たな値11は、書込が復号を伴うべきであることを特定する。復号を用いるべきであることを特定するのにどのコードまたはどの方法を用いるかは重要ではなく、またそれをオペコードで特定することすら重要ではない。重要なのは、任意の暗号化および復号が許されかつ示され、このため設計者がこの選択肢を利用できることのみである。図4dの実施例では2つの新たな構成ロジックレジスタが追加されている。アドレス1100および1101では、暗号ブロック連鎖（CBC）のスタータ値、および初期暗号鍵のためのアドレスを保持するためのレジスタが示される。

40

【0041】

任意の暗号化

この発明の別の局面に従うと、PLDはビットストリームの暗号化されたデータ部分および暗号化されていないデータ部分の両方を受入れることができる。ビットストリームの制御部分は、ビットストリームのデータ部分が暗号化されているかどうかを示す。ビットス

50

トリームのデータ部分が暗号化されていれば、これはPLD内で復号器へ迂回させられ、復号の後にPLDを構成する。データ部分が暗号化されていなければ、これは迂回させられず、PLDを構成するために直接用いられる。

【0042】

ビットストリームを暗号化しないことが好ましい場合がある。設計のデバッグ中に用いられる或る試験作業では、構成情報をリードバックすることが必須である。構成上の問題の診断は、(特に暗号化が問題と関わりを有しているかどうかを設計者が判断しようとする場合)暗号化のステップがまだ行なわれていない方がより単純となる。また、数人の設計者がPLDの複数部分で実現されるべきコードを書いており、PLDの異なった部分が異なったときに構成されるべきである場合、ビットストリームの部分すべてを可視にして、PLDを部分的に再構成可能にすることが必要であろう。

10

【0043】

図5aおよび図5bは、この発明の一実施例における暗号化されていないビットストリームと暗号化されたビットストリームとの違いを例示するために、まず暗号化されていない、次に暗号化された、同じ設計を表わす例示のビットストリーム部分を示す。実際のビットストリームは図の右に0および1を含み、左のテキストは含まない。左にあるテキストは右にあるビットの意味を説明するために入れたものである。これらビットストリーム部分は図4b~4dで例示したコマンドを用いる。図5aの暗号化されていないバージョンと、図5bの暗号化されたバージョンとの違いを強調するため、相違箇所を太字で示す。

【0044】

図5aを参照して、ダミーワード(すべて1として解釈される一定のハイの信号)および1と0との特定のパターンを有する同期ワード(sync word)の後、次のワードはタイプ001のものであり、10のオペコードを有し、000000000000100000のアドレスと、000000000001のワードカウントとを有する。したがってこのワードはコマンドレジスタCMDをアドレス指定し、1つのワードがここに書込まれるであろうことを特定する。図5aではビットストリームの左に注が付され、このワードがタイプ(Type)1であり、1つのワード(word)のCMDへの書込を指示することが示される。続くワード111はコマンドレジスタCMDに置かれるべきデータであり、CRC(巡回冗長検査)レジスタをリセットする。(好ましい実施例でPLDは、ビットストリームがロードされるのに伴いビットストリームからCRC値を算出するための、エリクソンによる米国特許第5,598,424号に記載のものなど図には示さない回路を含み、誤ったビットのロードを引起こしかねないビットストリーム電圧の不調から保護する。)次に、ヘッダワードによって、フォーマットはやはりタイプ1であり、フレーム長レジスタFLRに1ワードを書込むべきことを特定することが特定される。続くデータワード11001はフレーム長(25ワード)を特定する。同様に、いくつかの追加のヘッダおよびデータワードが続くが、ここにはフレームアドレスレジスタFARに書込むべきワードを特定するヘッダが含まれる。この場合、続くデータワードはデータがアドレス0で始まることを示す。最後に、これらレジスタがロードされた後、フレームデータ入力レジスタFDRIにデータを書込むためのコマンドが来るが、かなりのデータが書込まれることになるため、ワードカウントは000000000000として与えられ、タイプ2のヘッダによって、10530ワードがFDRIレジスタに書込まれることが特定される。これはPLDの構成を引起こす実際の設計データである。したがってビットストリーム内の次の10530ワードは設計データである。最後に、データが正しくロードされたことを確かにするために、構成データの出所であるデバイスが算出したCRC値がロードされ、PLDの算出したCRC値と比較される。追加のコマンドおよびデータがロードされ、こうして構成が完了したこと、およびPLDを動作モードに動かすべきことを示す。

20

30

40

【0045】

図5bは図5aと類似するが、データおよび注が太字で示されているところのみが異なる。図5bではデータは暗号化され、追加のコマンドを用いて初期鍵アドレスを与え、2つのワード(64ビット)をCBC(暗号化ブロック連鎖)レジスタに書込む。次に、タイ

50

プ1のヘッダはオペコード11を含み、データがフレームデータ入力レジスタFDR Iに書込まれる前に復号されるであろうことを示す。タイプ2ヘッダが続いてやはりオペコード11を有し、10530ワードが復号されデータ入力レジスタFDR Iに書込まれるべきであるという命令を与える。次に10530個の暗号化されたデータワードが続く。次に、(暗号化された)データが正しくロードされたことを確認するためのCRCワードが続く。最後に、追加のコマンドおよびデータが送られ、すべてが正しければPLDを動作モードに置く。

【0046】

復号プロセス

図6は、一実施例で任意の復号がどのように達成されるかを示す。図6は、構成ロジック29と、復号器24内に入るバス27、28とを詳細に示す。バス27は以下のものを含む。

- ・構成ロジック29内のレジスタアドレス1101(図4d)から取られる3ビットの初期復号鍵アドレス“Init_key_addr”、
- ・64ビットの、変更を加えられた暗号ブロック連鎖値“modCBC”。この値は、構成ロジック29内のレジスタアドレス1100(図4d)から取られる64ビットCBC値のより低いオードのビットを、レジスタ0001で特定される22ビットフレームアドレス値と取替えることによってもたらされる。
- ・ビットストリームから取られる、暗号化されたデータをロードするための64線“Encrypted_data”、
- ・復号器24が生成する復号されたデータを構成ロジック29に戻すための64線“Decrypted_data”、
- ・データが“Encrypted_data”線上にあり、復号器24が復号を始めることができることを復号器24に告げる信号“Enc_data_rdy”用の線、
- ・64ビットワードに対する復号が完了して“Decrypted_data”線上で利用可能であることを構成ロジック29に告げる信号“Dec_data_rdy”用の線、および
- ・たとえば鍵が或るセットの最初、中間または最後にあるべきかを指定する鍵メモリ内のビットによって特定される通りに鍵が用いられていない場合、構成ロジック29に構成を中断させてこれに従い状態レジスタをセットさせるために、復号器24により用いられるBad_key_set線。図4dに示す実施例では、状態レジスタはアドレス0111にあり、ビットのうち1つに論理1を記憶することによってBad_key_setエラーが示される。

【0047】

バス28は以下のものを含む。

- ・鍵アドレス用の3線。鍵アドレスは最初はバス27から与えられるものであるが、新たな鍵が用いられるたびに更新される。

【0048】

- ・復号鍵用の56線、および
- ・復号鍵が最初、中間、最後、または用いる唯一の鍵であることを示すための2線。

【0049】

設計再配置の防止

暗号化されたビットストリーム内の設計に対する1つの考えられる攻撃は、暗号化されたビットストリーム内でフレームアドレスレジスタ(開始アドレス)を変えることによって、これが復号されると、FPGAの使用中には可視であるFPGA部分へロードされるようにすることである。設計によってはブロックRAMの内容は可視である。入出力ポートの構成はあらゆる設計で可視であり、このため構成ビットは判断可能である。したがって、連続する設計部分がFPGAの可視部分へ動かされれば、FPGAが正しく機能しなかったとしても、攻撃者は再配置を繰返すことで、暗号化されていないビットストリームの内容を知ることができるであろう。

10

20

30

40

50

【0050】

設計再配置を防止するため、一実施例では、DES規格で用いられる暗号ブロック連鎖方法で使用される初期値に対して変更を加える。図7aおよび図7bはそれぞれこの発明に従い変更を加えられる三重DESアルゴリズムの暗号化部分および復号部分を示す。標準的な暗号ブロック連鎖方法は、開始数（これは設計者が与えるものであっても、またはランダムに生成されるものであってもよい）と、暗号化すべきデータの1番目のワードとのXORを取って暗号化プロセスを始める。この発明に従い、ランダムな数の一部はアドレス情報、すなわちこの例では、データが構成メモリ12にロードされることになる1番目のフレームの22ビットアドレスと取替えられる。64ビットの数であるスタートCBC値におけるxのラベルを付した最下位ビットは、yのラベルを付したフレームアドレスと取替えられ、こうして変更を加えられた64ビット値がもたらされるが、これはデータがロードされるアドレスに依存する。この変更を加えられたCBC値と、構成情報の1番目のワードであるワード1とのXORが取られる。次に暗号化アルゴリズムを用いて1番目の暗号化されたワードである暗号化ワード1をもたらし、これはビットストリーム内に置かれる。図7aは外部暗号ブロック連鎖を伴った三重暗号化アルゴリズムを示し、これは1番目の鍵を用いた暗号化ステップenc₁を含み、これに2番目の鍵を用いた復号ステップdec₂が続き、これに3番目の鍵を用いた暗号化ステップenc₃が続く。この1番目の暗号化されたワードである暗号化ワード1と、2番目の、暗号化されていないワードであるワード2とのXORが取られ、暗号化プロセスを繰返して暗号化されたワード2をもたらし。すべての構成データが暗号化されるまでXOR連鎖を継続する。

【0051】

図7bに示すように、PLDは逆のプロセスを行なって、復号されたワードを導くことになる。上記の暗号化の順番の場合、復号の順番は鍵3を用いた復号ステップdec₁、そして鍵2を用いた暗号化ステップenc₂、そして鍵1を用いた復号ステップdec₃となるであろう。重要なことに、復号ワード1を生成するための初期値の一部は暗号化および復号の両方に同じフレームアドレスを用いることになる。ビットストリームでなくPLDが、フレームアドレスレジスタに記憶されたフレームアドレスから、変更を加えられたCBC値を生成し、これはまた、構成データをロードすべき構成メモリ12のフレームを特定するためにも用いられる。したがって攻撃者が、データをロードすべきフレームアドレスを変えた場合、変更を加えられたCBC値はこれに従って変わり、構成データは正しく復号されない。

【0052】

XORステップは、暗号化の前に設計者のビットストリーム内にあった元のデータを生成する。たとえば元のデータ1＝復号ワード1となる。この復号された構成データはバス27（図3）で構成ロジック29へ送られる。

【0053】

構成ロジック29

構成ロジック29は、任意の暗号化をサポートするための構造と、設計再配置および単一の鍵攻撃を防ぐための構造とを含む。図6に示すように、構成ロジック29は保持レジスタ292、制御ロジック291、構成レジスタ（FDR1、FAR、CRC、およびinitCBCを図示する）、復号器24とのインターフェイス用マルチプレクサ294、295、64ビットのアセンブリレジスタ297、およびレジスタ298、299（構成アクセスポート21とのインターフェイス用）を含む。64ビットシフトレジスタ299はデータを構成アクセスポート21から受取り、このポートは1ビット幅のデータ用の単一のピンであっても、または8ビット幅のデータ用の8本のピンであってもよい。このデータは64ビットのシフトレジスタ299内にロードされ、これはレジスタ299が一杯になるまで行なわれる。次に、好ましくはこれら64ビットは64ビット転送レジスタ298内に並列にシフトされる。ここからマルチプレクサ296bは右および左の32ビットワードを交替で選択し、マルチプレクサ296aは制御線Mに制御されるのに応じて、一回につきデータの32ビットを、保持レジスタ292か、またはこれに代えてアセンブリ

10

20

30

40

50

レジスタ 297 のハイおよびローの部分のいずれかに送る。ビットストリームのロードが始まると、線 M および図示しないクロック信号は、マルチプレクサ 296 a および 296 b が 64 ビット転送レジスタ 298 から保持レジスタ 292 へデータを送ることを起こす。ここからこれらワードは制御ロジック 291 に与えられる。ワードがヘッダであれば、制御ロジック 291 はワードを解釈する。続くデータが暗号されずに書込まれるべきであることをオペコードが示していれば、制御ロジック 291 はアドレスをバス G に置いてレジスタを選択し、線 L 上に信号を置いてマルチプレクサ 294 がバス B をバス D に接続することを起こし、続くワードをバス B に与える。次のクロック信号（クロック信号は図示せず）の際、バス D 上のデータはアドレス指定されたレジスタにロードされる。図 4 d に示すレジスタすべてはこのようにロードされ得る。初期暗号ブロック連鎖値をロードするための *initCBC* レジスタは 64 ビットレジスタであり、図 5 b に示し先に論じたように 2 つの連続する 32 ビットワードを受取る。

10

【0054】

(1) *initCBC* レジスタに記憶された元の CBC 値と、(2) *FAR* レジスタに記憶された初期フレームアドレスとから得られる、変更を加えられた CBC 値は復号器 24 に利用可能である。一実施例では、*FAR* レジスタにある初期フレームアドレスが使用するのはいずれも 32 ビットであるのに対し、*initCBC* 値は 64 ビットを使用する。図 6 の実施例では、変更を加えられた CBC 値を与える 64 ビットバスは、フレームアドレスレジスタ *FAR* からの 22 ビットと、初期 CBC レジスタからの 42 ビットを含む。この発明が提供する安全性にとっては重要なことであるが、この値は構成データをロードするところに依存することに注目されたい。もし攻撃者が *FAR* レジスタの内容を変えることによって、暗号化されたデータを異なる場所にロードしようとするれば、復号器 24 に送り込まれる *modCBC* 値もまた変わることになる。

20

【0055】

或る数の構成データワードを復号するためのオペコードコマンドを制御ロジック 291 が受取ると、復号プロセスが始まる。制御線 M は、マルチプレクサ 296 a がデータを転送レジスタ 298 から、アセンブリレジスタ 297 へと通じるバス A に与えることを起こす。制御バス H は、暗号化されたデータのレジスタ 297 のハイ [31:0] およびロー [31:0] の部分にバス A を交替で接続し、こうして復号すべき 64 ビットワードが得られる。次に制御ロジック 291 は *Enc_data_rdy* 信号をアサートし、この信号は、復号器 24 がレジスタ 297 内のデータを復号することを起こす。

30

【0056】

復号器 24 は復号を行なうために、鍵アドレス *keyAddr* をバス 28 によって鍵メモリ 23 (図 3) に与える。これは、鍵メモリ 23 が、このアドレスにある 56 ビット鍵を 56 ビット *Key* 線に返すことを起こす。これはまた、鍵メモリ 23 が、やはりこのアドレスで鍵データに記憶されている 2 つの追加のビット “*Order*” を返すことを起こす。1 番目の復号鍵の場合、これら 2 つのビットはこれが 1 番目の鍵である、または唯一の鍵であることを示すことになる。もしそうでなければ、復号器 24 は *Bad_key_set* 信号をアサートし、この信号は制御ロジック 29 による構成動作の中断を起こす。もしこれら 2 つのビットによって、鍵が 1 番目の鍵または唯一の鍵であることが示されている場合、復号器 24 は、たとえば (シュナイヤーの同書に記載の) 周知の DES アルゴリズムを用いて復号を行なう。鍵が唯一の鍵でなければ、次に復号器 24 は鍵メモリ 23 における次のアドレスで鍵を得て、これが中間または最後の鍵であることを 2 つの *Order* ビットが示しているかどうかを調べる。もしそうでなければ *Bad_key_set* 信号がアサートされ、構成は中断される。もしそうであれば復号が行なわれる。もしこれが中間の鍵であれば次の復号作業が行なわれる。もしこれが最後の鍵であれば、復号器 24 は、復号されたワードと値 *modCBC* との XOR 関数をもたらし。次に復号器 24 は結果として得られた値を 64 ビット *Decrypted_data* バスに置き、*Dec_data_rdy* 信号をアサートする。この信号は制御ロジック 291 が信号を制御線 K に置くことを起こし、こうしてマルチプレクサ 295 が 64 ビットワードを 2 つ

40

50

のシークエンシャルな32ビットワードに分割することを起こす。制御ロジック291は信号を線Lに置いて、マルチプレクサ294が、復号されたデータの32ビットワードをバスDへ送ることを起こす。制御ロジック291はさらにバスG上にアドレス信号を置いてフレームデータ入力レジスタFDRIをアドレス指定する。次のクロック信号は復号されたデータをバスEに移し、ここでこれはフレームレジスタにロードされ、フレームレジスタが一杯であれば最終的に構成メモリ12内へ、FARレジスタで示されたアドレスにシフトされる。

【0057】

modCBC値は復号動作中ただ1回だけ用いられる。暗号化されたデータの後続の64ビットワードは、復号されてから、XOR演算のため先に復号されたデータを用いて連鎖される。(FARレジスタに記憶された値もまた、一度だけフレームアドレスの選択に用いられる。この後、フレームアドレスはフレームが一杯になるごとに単に増分される。)

【0058】

動作の流れ

図8は、構成ロジック29および復号器24により行なわれる動作の流れを示す。最初にステップ70において、構成ロジック29はビットストリームヘッダをロードし、対応するデータを図4bに示す構成ロジックレジスタに置き、これにはビットストリーム長の判断も含まれる。ステップ71で、準備シーケンスのさらなる部分として、構成ロジック29は1番目の構成メモリアドレスを読出す。ビットストリームフォーマットは、暗号化が用いられているかどうかを示すオペコードを含んでいることを想起されたい。ステップ72はオペコードの値による分岐である。暗号化が用いられていなければ、プロセスは図8の左部分に示されたものとなる。暗号化が用いられていなければ、プロセスは図8の右に示されたものとなる。暗号化がない場合、ステップ73で、構成ロジック29はカウンタをビットストリームワードカウント(図4cを参照)に等しく設定する。ステップ74で、構成データの32ビット(1ワード)が構成メモリ12のアドレス指定されたフレームへ送られる。ステップ75で、カウンタが終了していないことが示されれば、ステップ76でカウンタは減分され、構成データにおける次の1ワードが構成メモリ12に送られる。カウンタが終了していれば、構成ロジック29はクリーンアップ作業を行なうが、これには、最終的な循環冗長値を読出してビットストリームの終わりにある値と比較することによって、ビットストリームのロードにエラーがあったかどうかを判断することが含まれる。

【0059】

ステップ72で、ビットストリームが暗号化されていることが示されれば、カウンタにはワードカウントがロードされ、プロセスのステップ81で、鍵アドレスレジスタ293(図6)からの初期鍵アドレスが復号器24にロードされる。

【0060】

ステップ82で、暗号化された構成データの2つのワード(64ビット)が復号器24にロードされる。ステップ83で、アドレス指定された鍵が復号器24にロードされる。一実施例では、64ビットの数が復号器24にロードされる。この64ビットの数は、56ビットの鍵と、これが最初、中間、最後、または唯一の鍵であるかを示す2つのビットと、その他のビットとを含み、その他のビットは使用されないことも、パリティに使用されることも、または別の目的に使用されることもある。別の実施例では、64ビット鍵データは、これが最後の鍵であるかどうかを示す単一のビットを含む。さらに別の実施例では、64ビット鍵データは次の鍵についてのアドレスを含み、このため鍵を逐次的な順序で用いる必要はない。別の実施例では、付加的なビットがなく、鍵データが使用するビットは64個を下回る。さらに別の実施例では、鍵でなくビットストリームがいくつかの鍵を用いるべきかを示すが、これは安全性がより低いと考えられており、なぜなら攻撃者はいくつかの鍵が用いられているかを知って、一回につき1つの鍵を破る単一の鍵攻撃を行なうことができるからであり、これに対し、いくつかの鍵を用いるべきかを示すために鍵を用いれば、この情報が攻撃者に漏れることはない。

【0061】

10

20

30

40

50

ステップ84で、復号器24はたとえばDESアルゴリズムを用いて64ビットデータを56ビット鍵で復号する。DESアルゴリズムはブルース・シュナイヤーによる上述の書の265頁から278頁に記載されている。他の暗号化アルゴリズム、たとえば高度暗号規格AESを用いることもできる。他のアルゴリズムはより多くの鍵ビットを必要とすることがあり得る。たとえばAESは128ビットから256ビットの鍵を必要とする。

【0062】

ステップ85で、より多くの鍵を用いるべきかどうか判断される。鍵が最初、中間、最後、または唯一の鍵であることを示す2つのビットを調べて、これが最後の鍵であるかどうかを判断し、もしそうでなければ鍵アドレスは増分され、復号器24はメモリ23内で次の鍵をアドレス指定する。

10

【0063】

最後の鍵が用いられた後、ステップ87で、レジスタFARおよびinitCBCを組合わせて得られた64ビット値として図6に示した、変更を加えられたCBC値と、ステップ87で入手された復号された値とのXORが取られる。一実施例では、CBCレジスタ内にロードされた64ビットランダム数のうち22ビットは、ビットストリームの始めのフレームアドレスと交換される。暗号化プロセスの目的は、64ビットの暗号化された値におけるあらゆる2進数を、すべての先行のビットおよび鍵の関数にすることである。CBC値を1番目のアドレスと組合わせる目的は、意図された開始アドレスとは異なるアドレスにビットストリームがロードされると、暗号化された値が変わるようにすることである。ステップ87はこれら両方の目的を達成する。次に新たなCBCが記憶される。記憶は図6に示すFARおよびinitCBCレジスタであっても、または復号器24内に位置する別のレジスタであってもよい。

20

【0064】

ステップ88で、この復号された構成データはバス27(図3)で構成ロジック29へ送られる。構成ロジック29は更新された巡回冗長検査値を算出し、これはロードのプロセスの終わりにCRCレジスタに記憶された巡回冗長値と比較される。構成ロジック29が暗号化を使用するように設定されていれば、構成ロジック29内のマルチプレクサは、この復号された構成データを構成メモリ12のアドレス指定された列へ送る。

【0065】

ステップ89でカウンタが調べられ、もし終了していなければステップ96でカウンタは減分され、プロセスはステップ82へ戻り、ここで次の64ビット(2ワード)がビットストリームからロードされる。

30

【0066】

最後に、ステップ89でカウンタが終了したことが示されれば、ステップ90で、ビットストリーム内のCRC(巡回冗長検査)値が、ビットストリームのロードの際に算出されるCRC値と比較される。値が一致すれば、構成は完了しており、FPGAは動作へ移る。値が一致しなければ、ロードにエラーが発生しており、構成プロセス全体が中断される。

【0067】

鍵順序の評価—単一の鍵攻撃の防止

40

図9は、鍵順序を評価するために復号器24により実現される状態機械を示す。Enc_data_ready信号が活性化されるまで状態機械は状態S1に留まる。この信号は復号を始めてもよいことを示し、判断状態Q1へ移らせ、ここで復号器24は、バス27上のアドレスInit_key_addrにより特定されるアドレスをバス28に与え、鍵および鍵順序をリードバックし、さらに鍵順序データの2ビットから、鍵が1番目の鍵または唯一の鍵であるかどうかを判断する。もしそうでなければ、復号器24はBad_key_set信号を制御ロジック291に送り、構成ロジック29に構成を中断させる。アドレスが1番目または唯一であれば、復号器24は状態S3に移り、ここでデータが復号される。次に状態機械は判断状態Q2に移り、ここで鍵が最後または唯一であるかどうか判断される。もしそうであれば、復号は完了しており、状態S4で復号器24は復

50

号されたデータを構成ロジック 29 に返す。もしそうでなければ、状態 S5 で、復号器 24 は鍵アドレスを増分して新たな鍵を得る。状態機械は質問 Q3 を問い、次の鍵が中間の鍵または最後の鍵であるかどうかを判断する。もしそうでなければ、S2 は構成を中断させる。鍵が中間または最後であれば、状態機械は状態 S3 に戻り、データを再び復号する。別の実施例では、状態 S4 で復号器 24 はさらに、復号されたデータと C B C 値との X O R をとるステップを行なう。

【0068】

鍵順序を鍵内で記憶する利点は、攻撃者が単一の鍵攻撃を実行できないことであり、それは、復号器 24 が復号を行なう際に（設計者が意図したように）鍵メモリ 23 で特定された鍵すべてを使用することを、攻撃者は妨げることができないからである。単一の鍵攻撃を用いている攻撃者から保護するためには、2 番目および 3 番目の質問 Q2 および Q3 を問う必要はないが、それは鍵順序が P L D の内部にある鍵データ内に記憶されるからである。しかし、各々の鍵がロードの際に正しくラベル付けされたことを確かめるために、設計者、または鍵をロードするボードの検査員がこれら 3 つの質問を問うことは有益である。

10

【0069】

一実施例で復号器 24 は、復号—暗号化—復号の順番をとる三重 D E S 規格を用い、別の鍵が用いられるたびにアルゴリズムを（僅かだけ）交替させる。このような組合せは、A N S I X 9 . 5 2 1 9 9 8 三重 D E S 規格に準拠する。別の実施例ではその都度復号が用いられる。

20

【0070】

鍵メモリ 23

図 10 a に示す回路は 3 つの構成要素、すなわち電池電源スイッチ 22、制御ロジック 23 a および鍵レジスタ 23 b を含む。制御ロジック回路 23 a および鍵レジスタ 23 b は図 3 の鍵メモリ 23 を成す。図 10 a の実施例では、鍵レジスタ 23 b は 6 つの 64 ビットワードを含む。当然のことながら、これに代えて他の鍵メモリサイズを用いてもよい。実施例によっては、鍵メモリ 23 には 6 個をはるかに上回る鍵が記憶され、使用すべき鍵のアドレスを与えるためには 4 ビット以上が必要である。鍵レジスタ 23 b 用の電源は電池電源スイッチ 22 から線 V S W I T C H を通って来る。鍵メモリ電源電圧 V C C I が不十分、またはない場合、電池電源スイッチ 22 は電池バックアップ電圧 V B A T T を V S W I T C H 線に印加し、こうして V S W I T C H は正の電圧を伝える。

30

【0071】

この実施例では、各々の鍵レジスタは 64 個のメモリセルを有する。各セルは書込イネーブル信号 W E を受取り、この信号はハイであればセルへのデータの書込を引起こし、ローであればセル内のデータの保持を引起こす。書込イネーブル信号 W E は 1 レジスタ内のセルに共通である。P L D 電源電圧（V C C I とは異なる）がなく W E 信号が活性状態に駆動されない場合、T1 などの弱いプルダウントランジスタが W E 信号をプルダウンし、こうして鍵メモリレジスタをいずれもアドレス指定不可能にし、メモリセルのいずれも乱されないようにする。

40

【0072】

一実施例では、復号鍵を P L D 内にロードするために P L D の J T A G ポートを用いる。メモリセル電源電圧は通常の動作時には V C C I のデバイス電圧レベルにあり、一実施例でこのレベルは 3.0 ボルトから 3.6 ボルトの間である。J T A G ポートに与えられる信号はいくつかの異なった電圧であり得る。また、いくつかの異なった内部電圧があり得る。したがって電圧変位が必要とされる。この電圧変位はメモリセルで行なわれる。メモリセルの詳細を図 10 b で示す。インバータ I1 および I2 を含むラッチには、V S W I T C H で電力が与えられるため、デバイス電源電圧 V C C I があるかないかにかかわらず電力が与えられる。W E 信号および反転されたデータ信号 d a t a _ b の両方は 1.5 ボルトのレベルで動作する。これら信号は N M O S トランジスタ T4、T5 および T6 を駆動し、さらにインバータ I3 を通じて（やはり 1.5 ボルトの電源電圧を用いて）トラン

50

ジスタT7を駆動する。図10bは、WEがローであれば、トランジスタT4およびT5がオフであり、インバータI1およびI2を含むラッチの内容が維持されることを示している。WEがハイであれば、インバータI1およびI2のいずれかがローにされ、こうして新たなデータをラッチ内にロードする。

【0073】

制御ロジック回路23aはJTAGバス25（図3にも示す）から信号を受取る。JTAGバス25は、書込、読出、安全モードの設定のための制御信号、ならびにデータおよびアドレスバスを含む。このインターフェイスはIEEE1532JTAG規格に準拠する。鍵メモリ23がJTAGバス25によってアクセスされ得る前に、安全性状態（バス26）が非安全モードに置かれるが、これはISC__PROGRAM__SECURITY命令（図10aを参照）を用い、鍵データバスのビット0に論理1を与えることによって行なわれ得る。鍵メモリ23は、IEEE1532規格のISC__PROGRAMおよびISC__READ命令を用いて、JTAGバス25に書込まれ、およびここから（検証のために）読出される。制御ロジック23aは、JTAGバス25からの3ビットアドレス信号ADDRをデコードするためのデコーダを含み、JTAGバス25にISC__PROGRAM命令が現われれば書込ストロブ線ws__b[5:0]のうちアドレス指定された線上にローへのパルスを生じ、またはISC__READ命令がJTAGバス25上に現われれば読出選択線rsel[5:0]のうちアドレス指定された線上にハイの信号を生じする。6つの64ビットワードのうち1つの読出は、6本の読出選択線rsel[5:0]のうち1本にハイの信号を与えることで可能であり、これは読出マルチプレクサ23dが、64本の出力線q[63:0]上に選択されたワードを置くことを引き起こす。書込選択線または読出選択線のうち1本のみが一度に選択される。読出選択線がアサートされなければ、ハイのpark__low信号は、64個のトランジスタ23eが64線q[63:0]をプルダウンし、これら線のフローティングを防ぐことを引き起こす。

【0074】

鍵メモリ23が非安全モードで動作していれば、64ビットワードは鍵レジスタ23bからJTAGバス25へ読出され、ここで値はFPGAの外部で調べられ得る。FPGAはこの非安全モードにおいて、DES復号のための56ビット鍵としてレジスタ23b内の選択された64ビットワードのうち56ビットを用いることによって試験され得る。一実施例で、鍵メモリ23が非安全モードにあれば、設計がロード前に暗号化されていてもユーザの設計のリードバックは可能である。これによって設計者は、暗号化された設計も試験およびデバッグできる。鍵の安全性状態の通信はバス26（図3も参照）を通じて行なわれる。

【0075】

値が鍵レジスタ23b内に書込まれ、バス25からの読出動作で検証された後、ISC__PROGRAM__SECURITY命令を用いて、IEEE1532規格の一部である64ビット鍵データバスのビット0に論理0を与えることによって、制御ロジック23を安全モードに置く。この安全モードでは、鍵に対するアクセスは認められない。

【0076】

図11に示すように、攻撃者がISC__PROGRAM__SECURITY命令を用いてから鍵を読出すことによって非安全モードに戻ることを確実に防ぐために、安全性がなくなると（すなわちISC__PROGRAM__SECURITY信号が非安全論理レベルに移ると）制御ロジック23a内の状態機械は0を6つのワードすべてに、一回につき1ワードずつ書込むことで、すべての鍵を消去する。これは以下のステップ110からステップ119によって行なわれる。ステップ110で、0をwdata[63:0]バスに置き、ステップ111でws__b[0]信号を（論理0値で）アサートし、次にステップ112～117で、ws__b[0:0]からws__b[5:0]までの信号を一回につき1つずつ逐次的にストロブし、この後ステップ118で安全性状態を変えて非安全モードに入り、最後にステップ119でwdata[63:0]論理0値を解放する。このように、電池でバックアップされるメモリ23を非安全モードに置くあらゆる試みは、鍵レジ

スタ 2 3 b 内の値すべての消去を引起す。

【0077】

鍵メモリ 2 3 が安全モードにあるかどうかを通信するため、制御ロジック 2 3 a はバス 2 6 (これは単一の線であり得る) 上で安全モード信号を構成ロジック 2 9 に送り、鍵メモリ 2 3 が安全モードで動作していることを示す。この信号が非安全モードに切換わると、構成ロジック 2 9 は設計を構成メモリ 1 2 からクリアする。鍵が鍵レジスタ 2 3 b に記憶され、かつ鍵メモリ 2 3 が安全モードにあっても、暗号化されていないビットストリームが構成ロジック 2 9 により構成メモリ 1 2 内にロードされ得ることに注目されたい。

【0078】

鍵のロード、多数の暗号鍵

ユーザが設計の詳細を知ることができない安全モードに P L D が置かれる前に、復号鍵を P L D 内にロードする必要がある。図 3 に示す実施例では、鍵は J T A G ポート 2 0 を通じてロードされる。

【0079】

この発明の特徴として、暗号鍵はこの J T A G ポート 2 0 を通じてロードされる。J T A G プログラムがボード試験中に暗号鍵をロードすることが予測される。鍵を記憶するための R A M が非安全モードにあれば、ユーザはこれに対する完全なアクセスを有し、設計が暗号化されていたとしても鍵および設計の両方を読出すことができる。このことは鍵と鍵の使用とを試験している設計者にとって有用である。次に設計者は動作に一旦満足すると、J T A G ポートを通じ別の命令を送って鍵メモリを安全モードに置くことができる。鍵メモリが一旦安全モードに置かれると、鍵を読出すことはできない。さらに、鍵メモリを安全モードから非安全モードに移すと、メモリ初期化プロセスを開始する回路が活性化され鍵は消去される。(後に論じる図 1 5 で、この機能を行なうための状態機械を示す。)

【0080】

この発明の一局面に従うと、設計を暗号化するために 2 つ以上の鍵が用いられ得る。たとえば 3 つの鍵を用いる場合、ビットストリームはまず 1 番目の鍵を用いて暗号化され、次に結果として得られる暗号化されたビットストリームが 2 番目の鍵を用いて再び暗号化され、最後に、結果として得られる二重に暗号化されたビットストリームが 3 番目の鍵を用いてさらに暗号化される。この三重に暗号化されたビットストリームは、たとえば P L D を保持するプリント回路基板上の P R O M またはフラッシュメモリで記憶される。

【0081】

復号では、これら鍵を(逆の順序で)連続的に用いて、暗号化されたビットストリームを反復的に復号する。これに加え、特定の設計を復号するのに用いられるよりも多くの鍵が P L D に記憶されていれば、暗号化されたビットストリームは暗号化されていない部分の中に、いくつかの鍵を用いるべきかの指示と 1 番目の鍵のアドレスとを含み得る。このような実施例では攻撃者によるビットストリームの解読が容易となり得るが、それは攻撃者が一度に 1 つの鍵に対処するだけでよいからである。これに代えて、鍵自体が、最初、中間、最後、または唯一の鍵であることを示すこともある。こうして、同じ P L D が異なったときにプログラムされて、(異なった設計で構成される)異なった機能を行なうことが可能となり、異なった鍵の値についての情報を、設計者のうちただ一人または数人にとってのみ利用可能とできる。こうして第 1 の設計者は第 2 の設計について、両方の設計が同じ P L D で(異なったときに)実現されたとしても知らないことがあり得る。

【0082】

図 3 を参照して、構成ロジック 2 9 は図 1 の構成ロジック 1 4 の他に追加のロジックを含む。図 1 の構造では、構成アクセスポート 2 1 上のビットストリームはワード、一実施例では 3 2 ビットワードとして扱われる。これらワードのうち、通常はビットストリームの初めまたはその近くにあるいくつかのワードは、たとえばビットストリームの長さ、構成データについての開始アドレスなどのヘッダ情報を含む。この発明のビットストリームで新規な点は、ビットストリームが暗号化されているかどうかについての指示、およびビットストリーム内の構成データを復号するための鍵のアドレスである。

【0083】

電池でバックアップされるメモリ

鍵メモリ23に記憶される値は、FPGAへの電力が外されたときには電池によって維持されることが好ましい。

【0084】

さらに、スイッチ22などの電池電源スイッチを用いて、暗号鍵以外のメモリもまたバックアップできる。具体的に、PLDの電源切断時にPLDの生成したデータを保存することが目的であれば、PLD内のフリップフロップすべてにVSWITCH電圧供給が経路付けられるPLDを製造することができる。さらに、PLDの電源切断時にPLDの構成を保存することもまた目的であれば、構成メモリ12（図3）は代替的にVSWITCHから電力を与えられ得るが、このような実施例では、単にPLD内のフリップフロップに電力を与えるよりもかなり多くの電池電力が必要となり、さらにフリップフロップに電力を与えるには、数個の暗号鍵を記憶するための極めて小型のメモリに電力を与えるよりも多くの電池電力が必要となる。

10

【0085】

図12は電池電源スイッチ22の構造を示す。この実施例では、VBATTレベルシフト回路31によって、PLDが電池および主電源につき異なった電圧を用いることができる。さらに、当然のことであるが、この回路の目的はさまざまな電圧レベルに対処することである。一実施例で電池電源スイッチ22は最大3.6ボルトのVCCI電圧を扱うことができ、VCCIが約1ボルト未満に落ちると電池電力へ切換わる。電池電圧は1.0ボルトから3.6ボルトの間であり得る。

20

【0086】

電池電源スイッチ22は4個の出力駆動PチャネルトランジスタP0～P3を含む。トランジスタP0およびP1は一緒にオンおよびオフになり、トランジスタP2およびP3も同様である。この回路は、VCCIおよびVBATTが相互に接続されてしまうあらゆる可能性を回避するために、各々の脚につき1個でなく2個のトランジスタを含む。トランジスタP0は寄生ダイオード（ドレインと基板とのp-n接合）を含み、これはトランジスタがオフである際にも図で上方向へ電流を伝導するおそれがある。このような電流の流れを防ぐためにトランジスタP1が追加され、その基板はそのドレインに接続されるため、寄生ダイオードの伝導は下方向でのみ可能である。同様の仕組みをトランジスタP2およびP3にも設ける。したがって、電流がVBATTからVCCIへ、またはVCCIからVBATTへ伝導する可能性はない。インバータ33および34はVSWITCH電圧から電力を与えられるため、VCCIがオフであっても常に動作している。トランジスタP4は常にオンの抵抗器であり、静電放電に対する保護を提供する。ほとんどの時間、トランジスタP4により制御されるこれら構造は電流を引込まず、通常トランジスタP4にわたって電圧の降下はない。

30

【0087】

図13はVBATTレベルシフト回路31の一実施例を示す。端子OUTにおける出力電圧は信号INおよびINBにより制御される。これら信号はインバータ33および34により生成され、これらインバータはVSWITCHノードから自分の電源電圧を得る。したがって、VSWITCHがVBATTにより供給される場合、信号INおよびINBのうち一方は電圧VBATTにあり、他方は接地される。しかし、もしVSWITCHがVCCIにより供給される場合、INおよびINBのうち一方はVCCIの電圧レベルにあることになる。INがVCCIにあり、かつINBが接地されている場合、トランジスタ45はオンであり、トランジスタ46はオフである。Pチャネルトランジスタ43のゲートはローであり、トランジスタ43はオンであり、こうしてインバータ47の入力をVBATTにする。トランジスタ48の出力もまたVBATTにある。再び図12を参照して、トランジスタP0のゲートでの電圧レベルVBATTはトランジスタP0を正にオフにする。

40

【0088】

50

図14はVCCI検出回路32を示す。VCCI検出回路32は、線VSWITCHでの電圧がいつ電池に切換えられ、いつ再びVCCIに切換えられるのかを判断する。回路32のこの実施例は本質的に一続きになった5個のインバータ段I1からI5である。スイッチング電圧の制御は主にインバータ段I1で行なわれる。トランジスタ52および53はCMOSインバータを形成する。このCMOSインバータへの電力はPチャネルトランジスタ51を通じて流れることになり、トランジスタ51は、VCCIがトランジスタ51のしきい値電圧、典型的に0.7~0.8ボルトに達するまでオンにならない。VCCIのスイッチングが遅く、一杯の電圧に達するまで数ミリ秒かかる場合、トランジスタ51は回路I1の活性化を遅らせる。トランジスタ51がオンになると、トランジスタ52のソース（上の端子）はVCCIになる。Nチャネルトランジスタ53は典型的に、やはり約0.7~0.8ボルトのしきい値電圧を有するが、トランジスタ52に対して弱いトランジスタとしてのサイズにされる。一実施例でトランジスタ53の幅/長さ比は1/18であるのに対し、トランジスタ52の幅/長さ比は3/2である。したがってトランジスタ53は、トランジスタ52がオンになるまでインバータI2の入力をローにすぎない。一実施例で回路I1は、VCCIが約1.0ボルトにあるときにインバータ段I2の入力をハイにする。したがってインバータ54の出力はローになる。インバータ段I3はシュミットトリガである。インバータ段I3への0ボルトの入力はトランジスタ56および57をオフにし、トランジスタ55をオンにし、ノードN3をVCCIにし、トランジスタ58をオンにし、これはノードN4を引上げ、こうして、トランジスタ56がオンになる電圧を上昇させ、VCCIの小さな変動がノードN3での電圧を切換えることを防ぐ。インバータ59および60は任意であり、出力信号usebattおよびusebattbのより鋭いエッジをもたらし、これにより図12の電池電源スイッチ22がVBATTからVCCIに切換わることを引き起こす。VBATT'信号が制御するトランジスタ61は弱いプルダウンのトランジスタであり、VCCIがなくインバータ60から出力信号をもたらさないときにusebattb線を実際にローにする。

【0089】

構成されたPLDを含む製品の購入者に利用可能でない鍵

PLDを構成するのに用いられた設計を攻撃者が知ることを防ぐため、いくつかの追加のステップをとることがある。

【0090】

別の局面に従うと、PLDを組込んだシステムの販売前に鍵がPLD内にロードされ、これにより、PLDを含むシステムの販売後、設計をPLD内にロードして使用できるが、攻撃者は鍵に記憶された値を知ることにはできない。したがって、暗号化されていない設計は読出またはコピーされ得ない。この安全性を達成するためにいくつかのステップがとられる。

【0091】

安全モード保存（改ざん防止）

一実施例で、PLDの構成ロジック29には安全性に関する2つのフラグがある。一方は、復号鍵が安全にされているかどうかを示し、他方は、設計が復号された設計であり、したがって保護されなければならないことを示す。JTAGロジック13（図3）がISC__PROGRAM__SECURITY命令で安全モードを選択すると、制御ロジック23a（図10a）にあるsecure__keyフラグがセットされる。ビットストリーム内の設計データが暗号化されているということを、PLD内にロードされたビットストリームが示している場合、構成ロジック29（図示せず）にあるsecure__designフラグがセットされる。いずれかのフラグが後に非セットにされると構成メモリ全体がクリアされ、復号された設計がなくなると、secure__keyフラグが（ISC__PROGRAM__SECURITY命令により）リセットされると、鍵もまた消去される。

【0092】

図15は、設計クリア機能を行なう状態機械を示す。secure__designフラグがセットされると、状態機械は状態S1に入る。この状態は、secure__design

n フラグの安全モードから非安全モードへの変化を監視する。設計安全 (s e c u r e d e s i g n) モードが継続する限り状態機械は状態 S 1 に留まる。変化が一旦生じると、状態機械は状態 S 2 に入り、データを構成メモリ 1 2 内にシフトするためのデータシフトレジスタはリセットされ、こうして構成メモリビットについてのデータ線すべてに 0 を置く。次に状態機械は状態 S 3 に移り、アドレス指定されたフレームのワード線がアサートされる。この結果、データシフトレジスタ線上の 0 は、アドレス指定されたフレームにあるメモリビット内に書込まれる。質問 Q 1 で、アドレス指定すべきフレームがまだあることが示されると、状態機械は状態 S 4 に移り、ここでフレームアドレスが進められ、状態機械は状態 S 3 に戻る。質問 Q 1 で、アドレス指定すべきフレームがもうないことが示されると、プロセスは終了し構成メモリはクリアされる。

10

【0093】

鍵を攻撃者によるアクセスから保護することもまた必要である。鍵のロードは、設計を含むシステムが最終顧客にとって利用可能となる前に行なわれる。設計者は設計の開発中、PLD をデバッグのために非安全モードで動作させたいと望むことがある。このデバッグ動作を可能にし、かつ鍵の安全性を保存するため、鍵レジスタすべてのクリアで、鍵ロードのプロセスが非安全モードで始まる。鍵のロード中、および鍵が検証のためリードバックされる間、安全な鍵のフラグが非安全モードに保たれることになる。この安全鍵フラグはまた、構成ビットストリームがロードおよび復号される間も非安全モードに保たれる得る。しかし安全鍵フラグが一旦セットされると、安全鍵フラグを非安全モードに戻せばすべての鍵がクリアされ、さらに図 1 5 の状態機械の動作が開始される。したがって鍵がク

20

【0094】

リードバック攻撃、およびリードバックの禁止

いくつかの F P G A は F P G A からのビットストリームのリードバックを許しており、このためユーザは設計をデバッグしたり、または F P G A 内のフリップフロップから状態機械情報を入手できる。設計がリードバック動作用に再暗号化されない限り、ビットストリームをリードバックする行為によって、暗号化されていないビットストリームが可視となるよう暴露されてしまう恐れがある。

【0095】

設計のさらなる安全性は、暗号化された設計が F P G A にロードされる際にリードバックを禁止することで提供される。一実施例でリードバックは、復号鍵もまた安全にされる場合にのみ禁止される。

30

【0096】

図 1 6 は、構成メモリをロードおよびリードバックするための構造のブロック図を示す。一実施例で構成ロジック 2 9 は以下の 2 つの条件が揃っている場合にリードバックを防止する。すなわち、(1) データバス 2 6 (図 3 および図 1 0 を参照) 上の安全性状態線によって、鍵が安全モードにあることが示されている、および (2) ビットストリームが暗号化されていることを示す構成ビットストリーム内のオペコードに構成ロジック 2 9 が応答したことがある、の 2 つの条件である。したがって、鍵が安全にされていない、またはビットストリームが暗号化されていない場合、リードバックは許可され得る。別の実施例では、異なる条件によって、リードバックが許可され得るかどうかが制御される。

40

【0097】

構成ロジック 2 9 は、リードバックが行なわれるであろうことを示すヘッダをビットストリーム内で受取ると、そのフレームアドレスレジスタに記憶されたフレームアドレスを線 1 0 7 上に送り、このフレームアドレスはアドレスデコーダ 1 1 0 によってデコードされてバス 1 0 9 のアドレス指定された線を選択する。次に、線 1 0 8 上のワード線イネーブル信号がアサートされ、この信号はバス 1 0 9 の選択されたワード線をアサートし、こうして、選択されたワード線によりアドレス指定されたメモリセルは、その値を n 本のデータ線 1 0 2 上に置く (n はフレーム長であり、構成ロジック 2 9 内に記憶される)。次に構成ロジック 2 9 は線 1 0 4 上でロード信号をアサートし、データのフレームを (並列に

50

）データシフトレジスタ101内にロードする。次に構成ロジック29は線105上でシフト信号をアサートし、データシフトレジスタ101が32ビットワードのデータのフレームをバス103上でフレームデータ出力レジスタ（図4dを参照）へ、さらにそこから構成アクセスポート21（図3）上で出ていくビットストリームへシフトすることを起こす。

【0098】

ビットストリームに復号が示されている場合、構成ロジック29は内部フラグをセットしてこれを示す。これらフラグがセットされ、かつバス26上の安全性状態信号により示されるように鍵メモリ23が安全モードにあれば、構成ロジック29は、線108上のワード線イネーブル信号を非活性に保ち、かつ線104および105上のロードおよびシフト信号を非活性に保つことで、ビットストリーム内のリードバックコマンドに回答してリードバックを防ぐ。しかし鍵メモリ23が安全モードになれば、設計が暗号化されているかもしれないにもかかわらず、リードバックが許され、試験およびデバッグが可能となる。

10

【0099】

部分的再構成攻撃およびその防止

いくつかのFPGAはFPGAの部分的再構成を許すか、または、別個の開始アドレスおよび別個の書込命令を用いてFPGAの異なった部分に設計の異なった部分をロードすることを許している。攻撃者は設計を知る試みとして設計を部分的に再構成し、ブロックRAMまたはフリップフロップの内容を出力ポートに直接読出したり、または既存の設計に或るセクションを追加して、設計を知るのに用いられ得る情報を読出すことがあり得る。たとえば攻撃者は、暗号化されていない設計であって、その唯一の目的は暗号化された設計についての情報を抜き出すことである、暗号化されていない設計によって部分的にPLDを再構成することがあり得る。このようなトロイの木馬設計は別のビットストリームでPLD内にロードされたり、または既存の暗号化されたビットストリームに加えられたりする。たとえばFPGAのブロックRAM内にロードされた状態機械設計を知ることが攻撃者の関心であれば、トロイの木馬設計はブロックRAMのアドレスを通じて巡回し、ブロックRAMデータ内容をパッケージのピンに送るためのロジックを含み得る。

20

【0100】

攻撃者がこのような変化を加えることを防ぐため、元の設計が暗号化される場合、復号を伴う構成が一旦始まると構成ロジック29は部分的な再構成を却下する。構成ロジック29は、復号オペコードを有するヘッダが一旦処理されるとさらなる書込命令を却下する。さらに構成ロジック29は、暗号化を伴わない構成が一旦行なわれると、復号を伴う構成を却下する。構成ロジック29はこれらの制限を達成するために、復号命令が受取られた後に構成メモリに書込を行なうヘッダを無視し、さらに設計の暗号化されていない部分がロードされた場合に復号コマンドを有するヘッダを無視する。こうして、復号を伴う書込が用いられていることをオペコードのうちいずれかが示すと、PLDは単一の書込命令のみを受入れることになる。

30

【0101】

さらなる実施例

これら図面に関する以上の記載はいくつかの実施例についての詳細を与えている。しかし多くの追加的な実施例もまた可能である。たとえば、上述の暗号ブロック連鎖アルゴリズムの代わりに、たとえば一回につき1つの8ビットバイトなどの、ブロックサイズよりも小さなユニットでデータが暗号化され得る、暗号フィードバックモードと呼ばれる暗号化方法を用いることが可能である。この暗号フィードバックモードはシュナイダーの同書200頁から203頁に記載されている。

40

【0102】

さらに別の実施例では、暗号化が用いられると、あらゆるビットストリームはアドレス0で始まってロードされなければならない。この実施例の一実現例では、暗号化を特定するオペコードを受取ると、開始フレームアドレスレジスタFAR（図6）にロードされるア

50

ドレスをいずれもアドレス 0 と取替える。

【0103】

さらに別の実施例では、開始アドレスおよび設計データは両方とも暗号化される。この実施例では、暗号化されていない設計データで可能であると同様、異なったフレームアドレスで始まる暗号化された設計データのうちのいくつかのセグメントをロードすることが可能である。

【0104】

別の実施例では、鍵メモリ 23 などの鍵メモリに記憶される鍵データは、続く鍵の数を特定する。この実施例を変形したものでは、鍵データはさらにこの鍵に先行する鍵の数も特定する。設計者が意図した 1 番目の鍵アドレスとは違う鍵アドレスを攻撃者が与えると、構成は中断され得る。さらに、鍵内で特定された数の鍵が用いられるまで暗号化が進行する。

10

【0105】

別の実施例では、鍵メモリが非安全モードにある際に鍵をリードバック可能にする代わりに、鍵はパリティビットまたは CRC チェックビットを含み、これらビットのみが、鍵が正しくロードされたことを検証するためにリードバックされ得る。この実施例では、1 人の設計者が知っている鍵を別の設計者から秘密にしておくことができ、異なったときに PLD を用いて異なった設計をロードする際にこの実施例は有用である。

【0106】

上述の CRC チェックサム算出に関し、CRC チェックサムが設計の暗号化の前または後のいずれかに算出される実施例が提供され得る。当然のことながら、ビットストリームに追加されるチェックサムが設計データの暗号化前に算出される場合、その復号後に対応するチェックサムが設計データ上の PLD 内で算出されることになる。同様に、ビットストリームに追加されるチェックサムが設計データの暗号化後に算出される場合、PLD は設計データの復号前に、受取ったビットストリーム上で対応するチェックサムを算出することになる。

20

【0107】

復号鍵をロードするプロセスに関するさらなる注としては、図 8 に例示したプロセスを用いる場合、復号鍵をロードするためにデバイスプログラマを用いる必要はない。鍵は単にボード試験手順の一部としてロードされ得る。

30

【0108】

さらに、上述の構造および方法を用いて 2 つ以上の PLD をプログラムすることも可能である。いくつかのデバイスをデジタイゼーションに配置してこれら直列のデバイスを通じビットストリームを通過させるか、または直列のデバイスをアドレス指定することによって、単一のビットストリームを用いて 2 つ以上の PLD または FPGA をプログラムすることが周知である。デバイスのうち 1 つまたはそれ以上が暗号化された設計データを受取る際、いくつかの PLD をこのような配置に配置することが可能である。

【0109】

さらに別の実施例としては、暗号化された設計データを有するビットストリームにつき単一のアドレスのみが特定され得る一実施例を記載したが、別の実施例で、好ましくは暗号化されるいくつかのアドレスが、設計の別個の部分をロードするために特定され得る。さらにこれら別個の部分は同じ暗号鍵を用いても、異なった暗号鍵または異なった鍵の組を用いてもよい。

40

【0110】

以上の説明から明白となった変形例は、この発明の範囲内に含まれると意図されている。

【図面の簡単な説明】

【図 1】 先行技術の FPGA における機能関係を示す図である。

【図 2】 先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドを示す図である。

【図 3】 この発明の一実施例に従う FPGA での機能関係を示す図である。

50

【図 4】この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドを示す図である。

【図 5 a】暗号化されていないビットストリームの例を示す図である。

【図 5 b】暗号化されたビットストリームの例を示す図である。

【図 6】構成ロジック 29 と、復号器 24 に通じるバス 27 およびバス 28 の線とを示す図である。

【図 7 a】この発明の一実施例で用いられる三重暗号化を伴う外部暗号ブロック連鎖における、変更を加えられた開始値を示す図である。

【図 7 b】図 7 a とともに用いられる、対応する開始値および復号プロセスを示す図である。

10

【図 8】ビットストリームを処理するための動作を示すフローチャートである。

【図 9】鍵順序を評価するために復号器 24 により実現される状態機械を示す図である。

【図 10 a】図 3 の鍵メモリ 23 の構造を示す図である。

【図 10 b】図 10 a のメモリセルの構造を示す図である。

【図 11】非安全にされると鍵を消去するために図 10 a の制御ロジック 23 a が実行するステップを示す図である。

【図 12】図 10 a の電池電源スイッチをより詳細に示す図である。

【図 13】図 12 の電池電源スイッチのレベルシフト回路を示す図である。

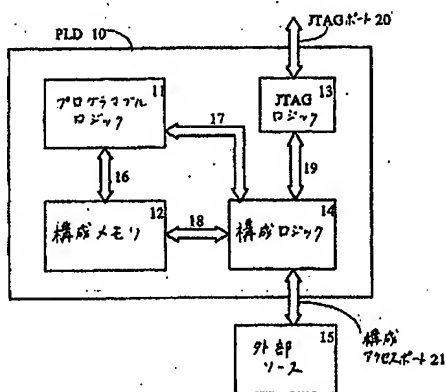
【図 14】図 12 の電池電源スイッチの電圧検出回路を示す図である。

【図 15】安全モードから出ると設計を消去するための状態機械を示す図である。

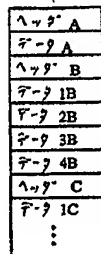
20

【図 16】構成メモリをロードし構成をリードバックするための要素を示すブロック図であって、暗号化があるときにディセーブルされる線を含む図である。

【図 1】

Fig. 1
PRIOR ART

【図 2 a】

Fig. 2a
PRIOR ART

【図 2 b】

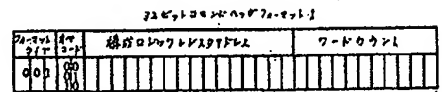


Fig. 2b PRIOR ART

【図 2 c】

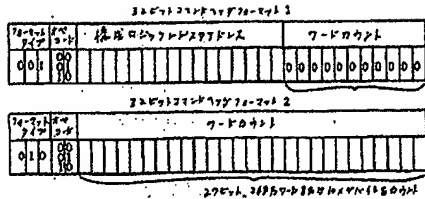


Fig. 2c PRIOR ART

【図 2 d】

ベニヤ・ローマ・ハッパ 標榜ロゴ・ レス・ワ・ド・ム	標榜ロゴ・ワ・レス・ム テ・ワ・肉名
0000	返田 元長 標章 (CRC)
0001	フレイム・アド・レス
0010	フレイム・デ・ワ・入
0011	フレイム・デ・ワ・出
0100	ワ・アド
0101	ワ・アド
0111	ワ・アド
1000	ワ・アド・ワ・アド
1001	ワ・アド・ワ・アド
1010	ワ・アド
1011	ワ・アド
1100	ワ・アド
1101	ワ・アド
1110	ワ・アド
1111	ワ・アド

Fig. 2d
PRIOR ART

【図 4 a】

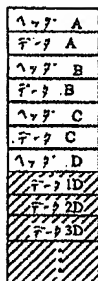


Fig. 4a

【 図 4 b 】

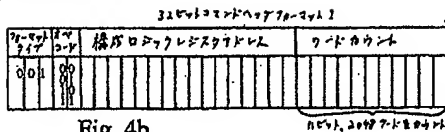


Fig. 4b

【図 3】

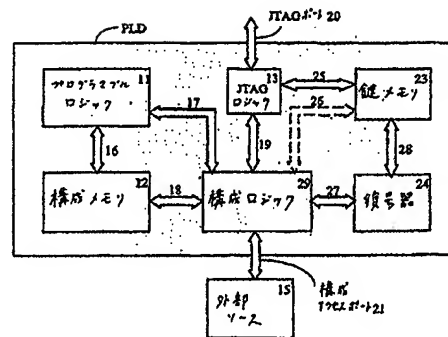


Fig. 3

【 図 4 c 】

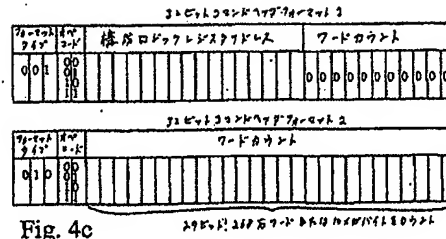


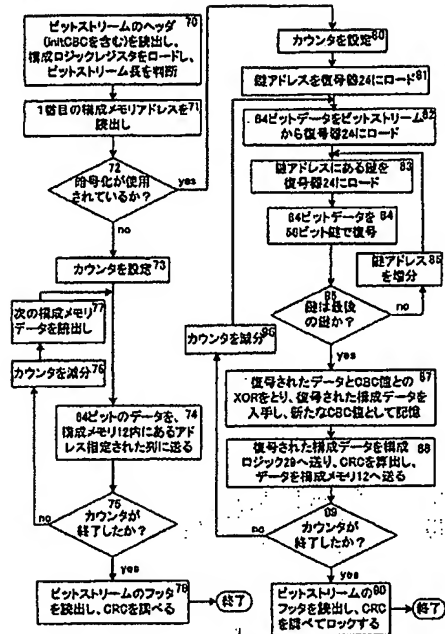
Fig. 4c

【 ㊦ 4 d 】

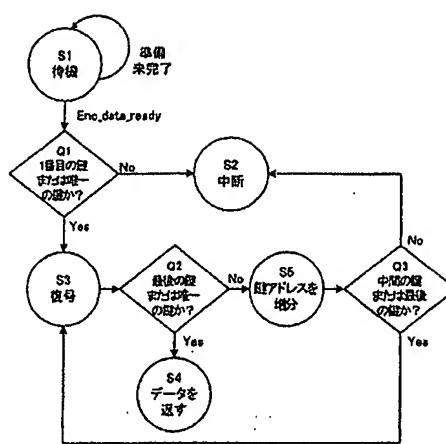
ピストン・スロー・フーズ	横濱ロジック・レジメ
横濱ロジック・レジメ	データ・内容
0000	巡回見直し検査 (C.R.C.)
0001	フレーム・アドレス
0010	フレーム・データ・入力
0011	フレーム・データ・出力
0100	コマンド
0110	制御
0111	状態
1000	マイジョー・イン・出力
1001	状態・マイジョー・イン
1010	マイジョー・出力
1011	マイジョー・入力
1101	明るさ・プロセッサ・速度 (C.R.C.) ストップ
1101	明るさ・プロセッサ・速度
1110	マイジョー・制御
1111	マイジョー・状態

Fig. 4d

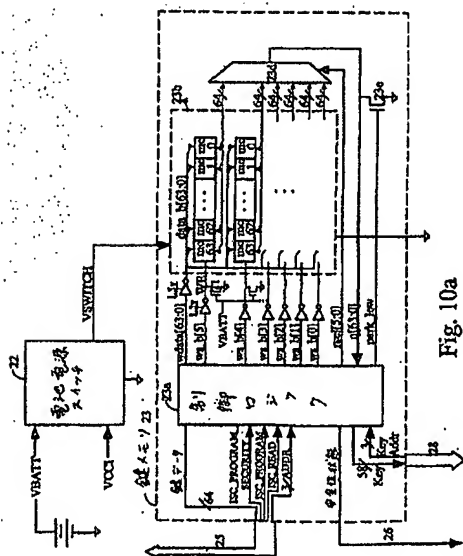
【図 8】



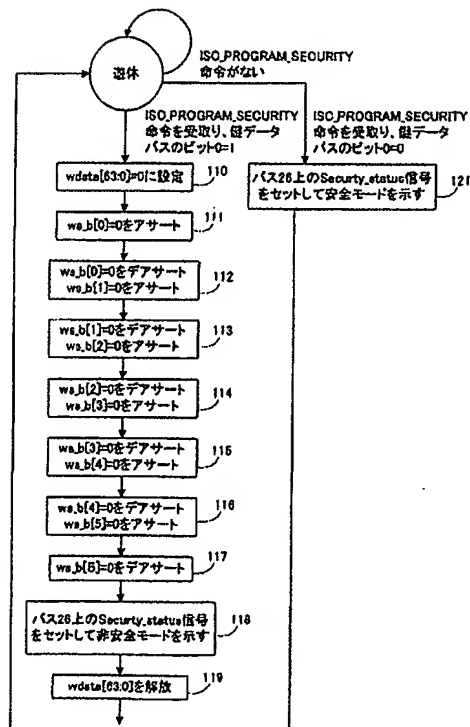
【図 9】



【図 10 a】



【図 11】



【図12】

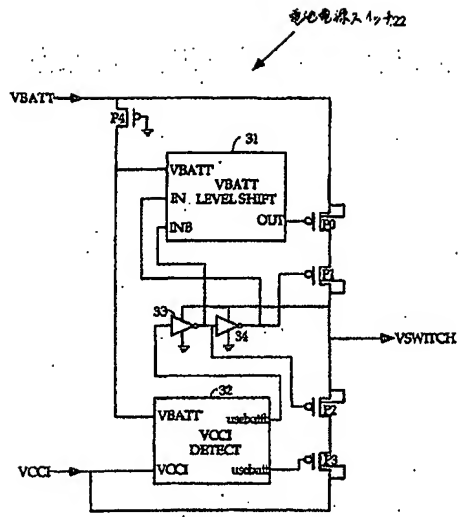


Fig. 12

【図13】

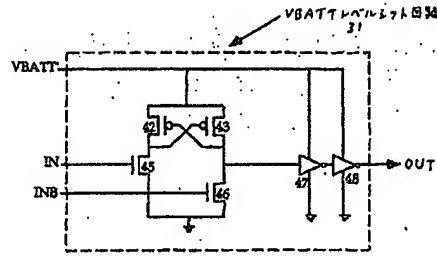


Fig. 13

【図14】

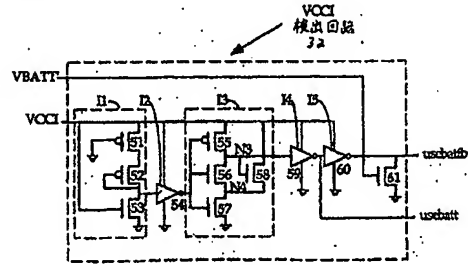
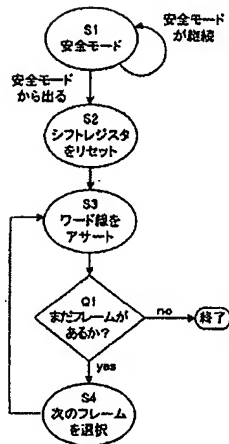


Fig. 14

【図15】



【図16】

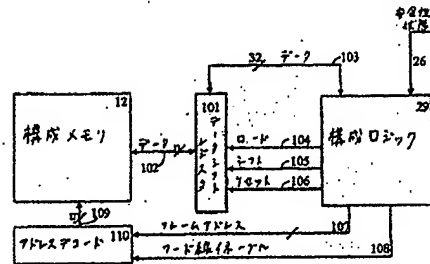


Fig. 16